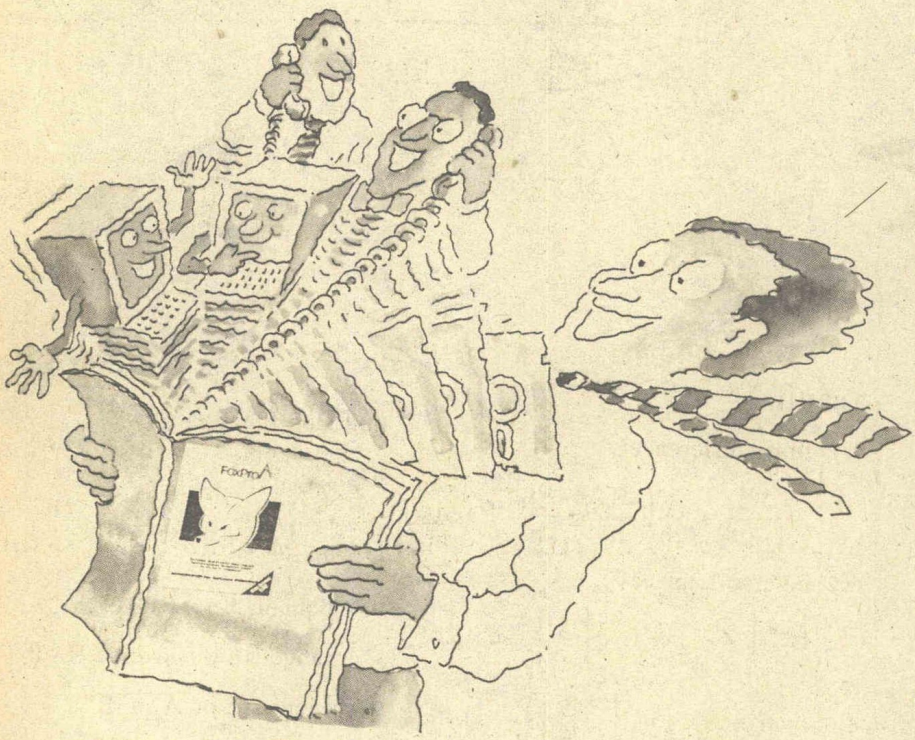
ประวิทย์ เรืองศาสตร์

ฟอกซ์โปร 2.0
จ้าวแห่ง DBMP (ตัวจริง)



ปรากฏตัวอย่างสง่าผ่าเผย
พร้อมด้วยอาวุธครบเครื่อง
เพิ่มพลังช่างรู้ใจผู้ใช้
เกมส์นี้ให้ฟอกซ์โปร 2.0

ถ้านับวันเดือนปีกันล่ะก็ จากข่าวคราวความคืบหน้าของการพัฒนาฟอกซ์โปรเวอร์ชัน 2.0 ที่เราได้รับทราบกันในวารสารไมโครคอมพิวเตอร์ ฉบับที่ 68 ไปกันแล้วนั้น ข่าวคราวของฟอกซ์โปรก็เงียบหายไปอย่างน่าใจหาย (โดยเฉพาะคนที่รอคอยอย่างผู้เขียนเองก็ซึ้งกับคำว่า "ความทรมานคือการรอคอย" เป็นอย่างดีทีเดียว) เมื่อเวลาได้ล่วงเลยมาถึงวันนี้ ฟอกซ์โปร 2.0 ก็ได้ปรากฏตัวออกมาแล้วอย่างสง่าผ่าเผยและแน่นอนที่สุด เมื่อผู้เขียนได้ชื่นชมฟอกซ์โปร 2.0 ไปพอสมควรแล้ว ก็รู้สึกคันไม้คันมือ อยากจะเล่าให้เพื่อน ๆ (ผู้เฝ้ารอคอยเหมือนกันกับผู้เขียน) ได้รับทราบและแบ่ง ๆ กันชื่นชมบ้าง หรือถ้าไม่ได้รอคอยก็ไม่เป็นไรครับ แต่ถ้าหากอยากจะรู้ว่าฟอกซ์โปร 2.0 จะดีเด่นเห็นชัดสักเพียงใด ก็ขอเชิญล้อมวงเข้ามาตรงนี้เลยครับ แล้วท่านจะพบว่าเวลาแห่งการรอคอยฟอกซ์โปร 2.0 ที่แสนยาวนานนั้น คุ้มยิ่งกว่าคุ้มอีกครับ

แกะกล่องและติดตั้ง

ฟอกซ์โปร 2.0 ที่ผู้เขียนได้มา 1 ชุด ประกอบด้วยแผ่นดิสค์ความจุ 1.2 เมกะไบต์ 4 แผ่น เมื่อได้มาก็รีบทำการติดตั้งก่อนอ่านคู่มือเสียอีก ผลปรากฏว่า ติดตั้งไม่สำเร็จในครั้งแรกเนื่องจากเนื้อที่ของฮาร์ดดิสค์ของผมเหลือน้อยเกินไป เลยต้องเปิดคู่มือซึ่งมีมาด้วย 6 เล่ม มีความหนารวมกันประมาณ 1 ฟุต ก็ได้ความว่าความจุเริ่มต้นของฮาร์ดดิสค์ที่ฟอกซ์โปรต้องการ สำหรับการติดตั้งคือ 6 เมกะไบต์ และถ้าต้องการติดตั้งส่วนเสริมและตัวอย่าง ก็จะใช้เนื้อที่อีก 6 เมกะไบต์และถ้าหากคุณได้ซื้อซอฟต์แวร์ส่วนที่เป็น Distribution Tool Kit มาด้วย ก็ต้องยอมเสียเนื้อที่อีกจำนวนหนึ่ง (ซึ่งผู้เขียนยังไม่ได้ซอฟต์แวร์ส่วนนี้มาเลยยังบอกไม่ได้) ส่วน

วิธีการติดตั้งก็ไม่ได้มีขั้นตอนสลับซับซ้อนยุ่งยากอะไรมากมาย แต่ถ้าจะให้ดีก็ควรจะมีคู่มือ แต่ก็รู้สึกเป็นห่วงและกลัวว่าท่านจะเจอปัญหาอย่างเดียวกันกับผู้เขียน (เนื้อที่ในฮาร์ดดิสค์ไม่พอ) จึงได้เตรียมทางเลือกในการติดเอาไว้ให้ดังนี้ครับ

1. Standard Verion + Help file ใช้เนื้อที่ 4.20 เมกะไบต์

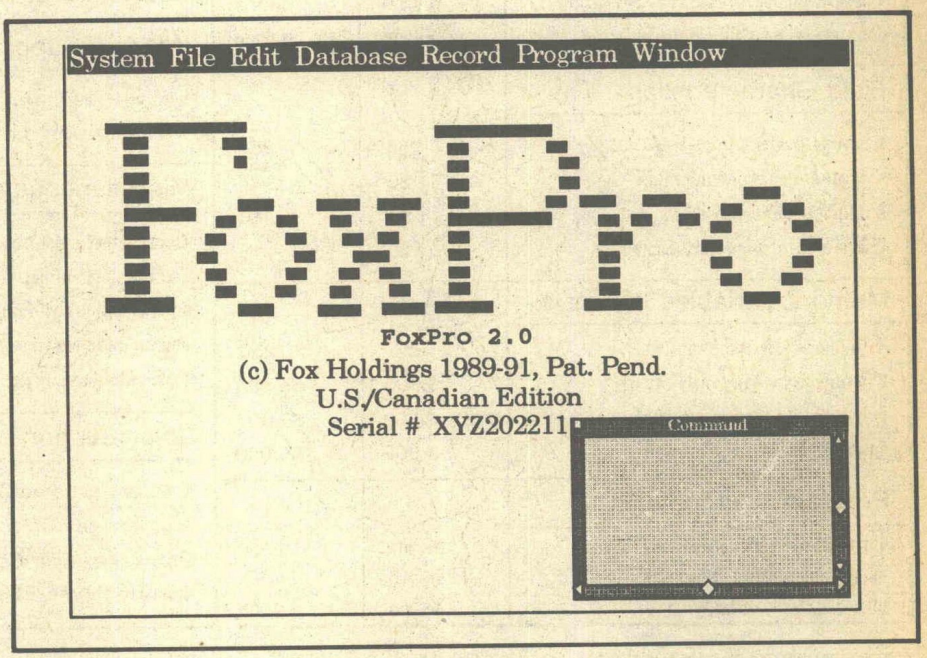
2. Standard Version + Help file + Extend Version ใช้เนื้อที่ 5.80 เมกะไบต์

3. ติดตั้งแบบสมบูรณ์ ซึ่งรวมทั้งไฟล์ตัวอย่างและส่วนเสริมต่าง ๆ ใช้เนื้อที่ 12.24 เมกะไบต์

แต่มีเรื่องสำคัญที่ผู้เขียนยังไม่ได้เรียนให้ทราบคือ เรื่องสเปคของระบบคอมพิวเตอร์ ที่ใช้ได้กับฟอกซ์โปร 2.0 ซึ่งได้รวบรวมไว้ดังรูปที่ 1

อุปกรณ์ที่ต้องการ	Standard	Extend	FoxPro/LAN
เครื่องคอมพิวเตอร์ตระกูล IBM หรือเครื่องคอมแพตทีใช้ CPU	8088/80286/80386 และ 80486	80386/80486	8088/80286/80386 และ 80486
หน่วยความจำที่ต้องการ	512 Kbyte	1.5 Mbyte	512 Kbyte
ระบบดำเนินงาน (DOS)	MS/PC DOS เวอร์ชัน 2.0 ขึ้นไป	MS/PC DOS เวอร์ชัน 2.0 ขึ้นไป	MS/PC DOS เวอร์ชัน 3.1 ขึ้นไป

รูปที่ 1 อุปกรณ์ที่ต้องการสำหรับการติดตั้ง/ใช้งานฟอกซ์โปร 2.0



รูปที่ 2 หน้าตาของฟอกซ์โปร 2.0

ไม่แต่งเติมแต่เพิ่มพลัง

เมื่อทำการติดตั้งเป็นที่เรียบร้อยแล้ว ผมก็รีบทดลองรันโปรแกรมทันที เพราะว่าอยากจะเห็นหน้าตาฟอกซ์โปร 2.0 เต็มทีแล้ว เมื่อรันโปรแกรมครั้งแรกก็ปรากฏว่า "หน้าตา" ของฟอกซ์โปรดังในรูปที่ 2 แทบจะไม่มีสิ่งใดเปลี่ยนแปลงไปจากรุ่นก่อน ๆ เลย (ถ้าหากคุณเคยใช้ฟอกซ์โปรมาก่อน) นอกจากข้อความที่อยู่ใต้โลโกเท่านั้น คาดคะเนเอาว่า ที่เป็นเช่นนี้คงจะเป็นความประสงค์ของผู้ผลิตฟอกซ์โปรคือ บริษัท ฟอกซ์-ซอฟต์แวร์ ที่ต้องการคงลักษณะของการติดต่อกับผู้ใช้ให้เหมือนเดิม ซึ่งทำให้ผู้ที่เคยใช้ฟอกซ์โปรรุ่นก่อน ๆ สามารถใช้งานต่อได้ทันทีและผู้ใช้ที่เริ่มใหม่ใช้ก็จะไม่รู้สึกอึดอัดจนเกินไป แต่ว่าความจริงแล้วมีความเปลี่ยนแปลงที่ดีขึ้นซ่อนรูปอยู่ข้างในมากมาย ถ้าหากคุณอยากจะสัมผัสกับความเปลี่ยนแปลงที่น่าตื่นตาตื่นใจ ก็ให้ทดลองรันไฟล์ DEMO.BAT ที่มีมากับฟอกซ์โปรลองดูก็แล้วกันครับ

เรื่องของตัวเลข

ความจริงแล้วผู้เขียนไม่ค่อยถนัดนักในเรื่องการเอาหาตัวเลขต่าง ๆ นานามาแสดง (และจะรู้สึกทำใจลำบาก ถ้าหากต้องทำตัวเลขขึ้นเอง) แต่ทว่าคราวนี้ไม่ลำบากนัก) เพราะว่าทางฟอกซ์-ซอฟต์แวร์ได้จัดเตรียมตัวเลขที่บ่งบอก "ความสามารถของฟอกซ์โปร 2.0" ไว้ให้แล้วเรียบร้อย ผู้เขียนเพียงแต่ทำหน้าที่ถ่ายทอดออกมาให้ผู้อ่านได้รับทราบกันดังที่เห็นในตารางรูปที่ 3

.EXE คอมพายล์เลอร์

การรอคอยฟอกซ์โปร 2.0 ของผู้เขียนและอีกหลาย ๆ คน อาจจะมีไฮไลต์ในจุดเดียวกัน คือพวกเราต่างรอคอยกันว่า ในการเขียนโปรแกรมในฟอกซ์โปร 2.0 โปรแกรมที่เขียนขึ้นนั้นจะสามารถคอมพายล์ให้เป็นไฟล์ที่มีส่วนขยายเป็น .EXE ซึ่งสามารถเรียกใช้ที่

ความสามารถของฟอกซ์โปร 2.0

Database & Indexes	Standard	Extend
จำนวนสูงสุดของเรคอร์ด/แฟ้มข้อมูล	1 พันล้าน	1 พันล้าน
จำนวนสูงสุดของตัวอักษร/เรคอร์ด	4,000	4,000
จำนวนสูงสุดของฟิลด์/เรคอร์ด	255	255
จำนวนสูงสุดของแฟ้มข้อมูลที่เปิดใช้/ครั้ง	25	25
จำนวนสูงสุดของตัวอักษร/ฟิลด์	254	254
จำนวนสูงสุดของความยาวคีย์ดัชนี.IDX	100	100
จำนวนสูงสุดของความยาวคีย์ดัชนี.CDX.	254	254
จำนวนสูงสุดของแฟ้มดัชนีที่เปิด/แฟ้มข้อมูล	ไม่จำกัด	ไม่จำกัด
จำนวนสูงสุดของแฟ้มดัชนีที่เปิดใช้ทั้งหมด	ไม่จำกัด	ไม่จำกัด
Field Characteristics		
จำนวนสูงสุดของตัวอักษร/ฟิลด์	254	254
จำนวนหลักสูงสุดของตัวเลข/ฟิลด์	20	20
ความยาวสูงสุดของชื่อฟิลด์	10	10
นัยสำคัญของการคำนวณทางตัวเลข	16	16
Memory Variables & Arrays		
ค่าปกติของตัวแปรหน่วยความจำ	256	256
จำนวนสูงสุดของตัวแปรหน่วยความจำ	3,600	65,000
จำนวนสูงสุดของตัวแปรแบบอะเรย์	3,600	65,000
จำนวนสูงสุดของสมาชิกในอะเรย์	3,600	65,000
Program & Procedure		
จำนวนบรรทัดคำสั่งสูงสุดต่อแฟ้มโปรแกรม	ไม่จำกัด	ไม่จำกัด
ขนาดสูงสุดของโปรแกรมคอมพายล์โมดูล	64 Kb	64 Kb
ขนาดสูงสุดของแฟ้มโพรซีเดอร์	ไม่จำกัด	ไม่จำกัด

Program & Procedure		
จำนวนคำสั่ง DO ที่ซ้อนกันได้	32	32
จำนวนคำสั่ง READ ที่ซ้อนกันได้	5	5
จำนวนสูงสุดคำสั่งที่เป็นโครงสร้าง	64	64
Report Writer Capacities		
จำนวนสูงสุดขององค์ประกอบในรายงาน	ไม่จำกัด	ไม่จำกัด
จำนวนบรรทัดสูงสุดในการกำหนดรายงาน	255	255
จำนวนสูงสุดของการจัดกลุ่มข้อมูล	20	20
Window Support		
จำนวนสูงสุดของวินโดว์ที่เปิดใช้งาน	ไม่จำกัด	ไม่จำกัด
จำนวนสูงสุดของบราวส์วินโดว์	25	25
Miscellaneous Capacities		
จำนวนสูงสุดของตัวอักษร/ตัวแปรแบบสตริง	64Kb	2 จิกะไบต์
จำนวนสูงสุดของอักษร/บรรทัดคำสั่ง	2,048	2048
จำนวนสูงสุดของไฟล์ที่เปิดได้	99	จำกัดที่ DOS
จำนวนสูงสุดของการกดคีย์บอร์ด/มาโคร	1024	1024
จำนวนฟิลด์สูงสุด/คำสั่ง SELECT (SQL)	255	255
Color Support		
จำนวนของ scheme/Color Set	24	24
จำนวนสูงสุดของ Color Set	ไม่จำกัด	ไม่จำกัด
จำนวนสูงสุดของสีต่อ Color Scheme	10	10
จำนวน Scheme ที่ให้ผู้ใช้เปลี่ยนสีได้	8	8

รูปที่ 3 ความสามารถของฟอกซ์โปร 2.0

DOS โดยไม่ต้องพึ่งพาฟอกซ์โปรอีกเลย ได้หรือไม่? คำตอบก็คือ ได้แล้ว... แต่ว่า อย่าเพิ่งไชโยนะครับ เพราะว่าคุณจะต้องไปจัดหาซอฟต์แวร์ส่วนที่เป็น Distribution Took Kit (ซึ่งใช้สำหรับเรื่องนี้โดยตรง) มาเสียก่อน สำหรับข้อมูลที่ทราบมา (ผู้เขียนยังไม่ได้ทดลองเพราะยังหาซอฟต์แวร์ชุดนี้ไม่ได้) คุณสามารถเลือกสร้างโปรแกรมได้ 2 แบบ คือ

1. Compact Executeable File คือโปรแกรมที่มีส่วนขยายเป็น .EXE แบบกระชับโดยขนาดของไฟล์จะเล็กมาก แต่ว่าต้องใช้งานร่วมกับไลบรารีไฟล์ที่มีขนาด 1.2 เมกะไบต์

2. Stand Alone Executeable File คือโปรแกรมที่มีส่วนขยายเป็น .EXE ที่สามารถรันได้ด้วยตัวเองอย่างอิสระ แต่ก็มีเรื่องน่าเศร้าอยู่นิดหนึ่งตรงที่ไฟล์แบบนี้มีขนาดใหญ่มาก โดยจะเริ่มต้นที่ขนาด 656 กิโลไบต์ (แต่ถ้ามองกันในแง่ของความเป็นอิสระต่อเรื่องลิขสิทธิ์ของโปรแกรมที่เราสร้างขึ้นแล้ว ถึงแม้ว่าไฟล์ที่ได้จะมีขนาดใหญ่ไปหน่อย แต่ก็ดีกว่าทำไม่ได้) และถ้าจะให้ดีจริง ๆ แล้ว คุณก็น่าจะจัดหาซอฟต์แวร์ API/-Construction Kit ซึ่งใช้ในการเชื่อมโยงโปรแกรมของฟอกซ์โปรกับโปรแกรมภาษาซีหรือแอสเซมบลีมาด้วย ก็จะทำให้คุณมีเครื่องไม้เครื่องมือที่ครบถ้วนสมบูรณ์ พร้อมที่จะทำโปรแกรมได้ทุกรูปแบบและสภาวการณ์

ความเร็ว สุด ๆ ไปเลย

เรื่องของความรวดเร็วก็ออกจะเป็นเรื่องที่ผู้ใช้ DBMP ทุก ๆ คนปรารถนา เมื่อย้อนความหลังแล้วก็จะเห็นว่า DBMP ตระกูลฟอกซ์ (ฟอกซ์เบสและฟอกซ์โปร) ได้นำหน้าในเรื่องความเร็วมาโดยตลอด

แต่มาคราวนี้ฟอกซ์โปร 2.0 ออกจะล้ำหน้ามากขึ้นไปอีก เพราะว่าจากการที่ฟอกซ์โปรได้ใช้เทคนิคในการเข้าถึงข้อมูลที่มีชื่อว่า Rushmore Technology ซึ่งเป็นเทคนิคใหม่ล่าสุดของฟอกซ์โปรเอง ทำให้ประสิทธิภาพของการเข้าถึงข้อมูลของฟอกซ์โปร 2.0 ก้าวล้ำนำหน้า DBMP (บนไมโครคอมพิวเตอร์) ตัวอื่น ๆ อยากชนิดหารอยเท้าไม่พบเลยทีเดียว จากการทดสอบของฟอกซ์-ซอฟต์แวร์เอง เมื่อทำการทดสอบแข่งกับโปรแกรม DB2 บนเครื่องระดับเมนเฟรม โดยใช้ฟอกซ์โปรที่รันบนเครื่องพีซี (ไม่แน่ใจว่าเป็นเครื่องที่ใช้ซีพียู 80486 หรือเปล่าเพราะเขาไม่ได้บอกไว้) ในการสอบถามข้อมูลที่มีเงื่อนไขซับซ้อนจากแฟ้มข้อมูลขนาด 200 เมกะไบต์ที่มีรายการข้อมูลถึง 1,000,000 เรคอร์ด ผลจากการทดสอบก็ปรากฏว่าฟอกซ์โปรมีความเร็วสูงกว่าถึง 4 เท่า และในคู่มือยังกล่าวไว้อีกว่า ถ้าทดสอบกับ DBMP บนเครื่องพีซีด้วยกันแล้วฟอกซ์โปรจะทำได้เร็วกว่าเป็นร้อยเท่าเลยทีเดียว แทบไม่น่าเชื่อเลยใช่ไหมครับ แต่จุดที่พอจะกล่าวสนับสนุนการทดสอบนี้ได้ก็คือ ในการสอบถามข้อมูลโดยมีเงื่อนไขที่ซับซ้อนนั้น ฟอกซ์โปร 2.0 สามารถให้ผลการทดสอบที่ดีเลิศก็เพราะว่า Rushmore Technology ที่ฟอกซ์-ซอฟต์แวร์สร้างขึ้น มีจุดเด่นในเรื่องนี้โดยเฉพาะ ดังนั้นเมื่อเปรียบ DBMP ตัวอื่น ๆ จึงทำให้เกิดผลความแตกต่างอย่างมากมาย แต่อย่างไรก็ตามผู้เขียนก็กำลังรอให้ห้องปฏิบัติการทดสอบ (ที่วางตัวเป็นกลาง) ทำการทดสอบและรายงานผลออกมาอย่างชัดเจนเสียก่อน ซึ่งจะทำให้ผลการทดสอบดูเป็นกลางและถูกต้องยุติธรรมขึ้น

แฟ้มดัชนีที่ฝันถึง

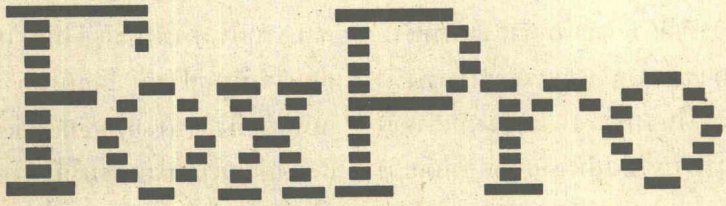
ขออนุญาตสมมติว่าท่านคุ้นเคยกับเรื่องการสร้างแฟ้มดัชนี (Index) กันมาบ้างแล้ว จึงขออธิบายเพียงสั้น ๆ ว่าแฟ้มดัชนีคือ แฟ้มที่ทำหน้าที่ควบคุมลำดับของรายการข้อมูลให้เป็นไปตามนิพจน์ที่กำหนดขึ้นเมื่อทำการสร้างแฟ้ม-ดัชนี ความก้าวหน้าของการทำดัชนีของฟอกซ์โปร 2.0 จัดเป็นความก้าวหน้าแบบก้าวกระโดดเลยทีเดียว เพราะแต่ก่อนนี้ในฟอกซ์โปรจะมีการสร้างแฟ้มดัชนีได้แบบเดียว ซึ่งเมื่อสร้างแล้วก็จะได้แฟ้มดัชนีที่มีส่วนขยายเป็น .IDX แต่ในเวอร์ชัน 2.0 เราสามารถสร้างแฟ้มดัชนีได้ถึงสองแบบคือ แบบคีย์เดี่ยว (Single Entry Index) และแบบคีย์รวม (Compound Index) จุดเด่นของเรื่องนี้จะอยู่ที่แฟ้มดัชนีแบบคีย์รวม เพราะสามารถสร้างแฟ้มดัชนีให้มีคีย์สำหรับควบคุมข้อมูลได้หลาย ๆ คีย์อยู่ในแฟ้ม-ดัชนีเดียวกัน ซึ่งมีประโยชน์หลายอย่างสำหรับการทำโปรแกรม เป็นต้นว่า ลดจำนวนแฟ้มดัชนีและยังช่วยให้สามารถจัดการเรื่องของดัชนีข้อมูลได้อย่างคล่องตัวอีกด้วย นอกจากนี้แล้วยังมีการกำหนดลักษณะพิเศษของแฟ้มดัชนีได้อีกสามแบบคือ Compact, Structural, Non-Compact โดยแต่ละแบบก็จะมีคุณสมบัติเฉพาะตัวดังนี้

– แฟ้มดัชนีแบบกระชับ (Compact Index) แฟ้มดัชนีแบบนี้มีจุดเด่นมากตรงที่กินเนื้อที่ของดิสก์น้อยมาก โดยใช้เนื้อที่เพียงหนึ่งในหกของแฟ้มดัชนีแบบเดิมในฟอกซ์เบสพลัสและฟอกซ์โปร 1.XX ซึ่งส่งผลให้การเข้าถึงข้อมูลโดยใช้แฟ้มดัชนีแบบนี้ มีความเร็วที่ดีขึ้นอีกด้วย

– แฟ้มดัชนีแบบเดิม (Non-Compact Index) สำหรับแบบเดิมนี้เรียกได้อีกอย่างว่าเป็นแบบไม่กระชับ ซึ่งคุณอาจจะสงสัยว่าแล้วมีไว้ทำไมกัน เพราะว่าถ้าหากท่านต้องการใช้ข้อมูลร่วมกับโปรแกรมที่เขียนด้วย ฟอกซ์โปรรุ่นก่อน ๆ นี้หรือฟอกซ์เบสพลัส ก็ต้องสร้างแฟ้มดัชนีด้วยการระบุให้เป็นแบบไม่กระชับนี่แหละครับ ถึงจะใช้งานร่วมกันได้

– แฟ้มดัชนีแบบโครงสร้าง (Structural Index) แฟ้มดัชนีแบบนี้จะมีชื่อเหมือนกับแฟ้มข้อมูลที่เป็นคู่กันอยู่และเมื่อมีการเปิดแฟ้มข้อมูล แฟ้มดัชนีแบบนี้ก็จะถูกเปิดขึ้นโดยอัตโนมัติ

ความพิเศษของการทำดัชนีอีกอย่างหนึ่งคือ คุณสามารถกำหนดการเรียงลำดับของคีย์ได้สองแบบคือ เรียงจากน้อยไปมาก (Ascending Order) และมากไปน้อย (Descending Order) และมีอยู่เรื่องหนึ่งซึ่งผู้เขียนรู้สึกประทับใจมากคือ การที่สามารถใช้ฟังก์ชันที่เราสร้างขึ้นเอง (UDF : User-Define Function) มาสร้างนิพจน์สำหรับการทำดัชนีได้ ซึ่งตรงจุดนี้แหละจะทำให้เรา



FoxPro 2.0

สามารถสร้างแฟ้มดัชนีที่มีการเรียงลำดับของข้อมูลแบบตัวอักษรได้อย่างถูกต้องตามหลักของการจัดเรียงภาษาไทยของพจนานุกรมไทย (ถือเอาฉบับราชบัณฑิตยสถานเป็นมาตรฐาน) ซึ่งแต่เดิมนั้นแฟ้มดัชนียังจัดลำดับของรายการตามน้ำหนักของค่ารหัสแอสกีซึ่งไม่ถูกต้อง แต่คราวนี้ก็หมดปัญหาแล้วล่ะครับ แต่ก็ต้องขอเรียนให้ทราบไว้ก่อนว่า ตรงจุดนี้เป็นเพียงแนวความคิดของผู้เขียนเท่านั้นนะครับ (ซึ่งเห็นว่าน่าจะทำได้) แต่ในทางปฏิบัติแล้วจะพยายามนำเสนออีกครั้งยัง...ไม่หมดแค่นั้นครับ จะไม่ดีแน่ถ้ากล่าวถึงเรื่องการทำดัชนีแล้วไม่กล่าวถึงเรื่อง การสร้างความสัมพันธ์ระหว่างแฟ้มข้อมูลโดยในฟอกซ์โปร 2.0 มีคำสั่งที่ช่วยในการสร้างความสัมพันธ์ระหว่างแฟ้มข้อมูลแบบหนึ่งต่อหนึ่ง (One to One) อยู่แล้ว โดยการใช้คำสั่ง SET RELATION TO <Expr> INTO <alias> แต่ในบางกรณีที่เราต้องการสร้างความสัมพันธ์ระหว่างแฟ้มข้อมูลให้เป็นแบบหนึ่งต่อมากกว่า (One to Many) เราก็สามารถทำได้โดยการใช้คำสั่ง SET SKIP TO [<alias>, <alias>] ส่วนในรายละเอียดการใช้งานนั้นคิดว่าคงต้องให้คุณทดลองใช้จริง ๆ ดูก่อน แล้วคุณจะเห็นประโยชน์และถึงบางอ้อได้ด้วยตัวเองจึงของดกล่าวในที่นี้ครับผม

อะเรย์ที่เก่งกาจ

สำหรับผู้ที่เคยเขียนโปรแกรมมาแล้ว คงจะรู้จักกับตัวแปรแบบอะเรย์กันพอสมควร สำหรับผู้ใช้มือใหม่ตอนนี้ขอสมมติว่าคุณรู้จักกับตัวแปรอะเรย์ก่อนก็แล้วกัน และเมื่อพูดถึงอะเรย์แล้วทำให้จำเป็นต้องเปรียบเทียบกับโปรแกรมคลิปเปอร์ 5.0 เพราะว่าในเรื่องอะเรย์แล้วโปรแกรมคลิปเปอร์มีฝีมืออยู่ในระดับแชมป์ เมื่อดาวรุ่งอย่างฟอกซ์โปร 2.0 ได้ทำการพัฒนาเรื่องการจัดการอะเรย์ขึ้นมา ก็ต้องประกบคู่กันหน่อย และแล้ว ...ผลการให้คะแนนก็ออกมา ก็ปรากฏว่ากรรมการทั้งสามท่านให้คลิปเปอร์ชนะไปโดยเอกฉันท์ แต่ว่าฟอกซ์โปรก็ทำได้ดีพอสมควรทีเดียว โดยเพิ่มฟังก์ชันสำหรับการจัดการข้อมูลในอะเรย์ขึ้นมาดังที่ได้แสดงไว้ในรูปที่ 4 และจุดที่เข้าตากรรมการมากก็มีอยู่หลายจุดอยู่ทีเดียว เป็นต้นว่าในคำสั่ง SCATTER และ GATHER นั้น ทำให้สามารถจัดการส่งข้อมูลไปมาระหว่างอะเรย์กับเรคอร์ดในแฟ้มข้อมูลเป็นไปอย่างสะดวกสบาย และการประกาศใช้อะเรย์ก็สามารถกำหนดได้ว่าต้องการให้เป็นสาธารณะหรือเฉพาะที่ การปรับเปลี่ยนขนาดของอะเรย์ก็ทำได้โดยสะดวก และจุดที่ผู้เขียนชอบมากก็เห็นจะเป็นจุดที่ฟอกซ์โปรสามารถผ่านค่าทั้งอะเรย์ไปยังโปรแกรมย่อยได้เป็นอย่างดี โดยเป็นการผ่านค่าแบบอ้างอิง (Pass by Refference) ดังได้แสดงตัวอย่างโปรแกรมไว้ในรูปที่ 5 จะเห็นว่าต้องใช้เครื่องหมาย '@' นำหน้า

ฟังก์ชัน	ความหมาย
ADEL( )	ลบสมาชิกออกจากอะเรย์
ADIR( )	นำเอาชื่อไฟล์ที่ระบุเก็บเข้าอะเรย์
AELEMENT( )	ให้ค่าหมายเลขสมาชิกจากการผ่านค่า Row, COl
AFIELDS( )	นำเอาโครงสร้างแฟ้มข้อมูลมาเก็บในอะเรย์
AINS( )	เพิ่มจำนวนสมาชิกในอะเรย์
ALEN( )	ให้ค่าจำนวนสมาชิกในอะเรย์
ASCAN( )	ค้นหาค่าสมาชิกในอะเรย์จากนิพจน์ที่ระบุในฟังก์ชัน
ASORT( )	จัดเรียงค่าสมาชิกในอะเรย์
ASUBSCRIPT( )	ให้ค่า Row, Column จากการผ่านค่าหมายเลขสมาชิก

รูปที่ 4 คำสั่งและฟังก์ชันที่น่าสนใจเกี่ยวกับอะเรย์

```
DIMENSION multiply(3)              && สร้างอะเรย์
STORE 2 TO multiply(1)             && กำหนดเลขตัวตั้ง
STORE 4 TO multiply(2)             && กำหนดเลขตัวคูณ
STORE 0 TO multiply(3)             && ผลลัพธ์
= product (@multiply)              && ทำฟังก์ชันโดยผ่านค่าไปทั้งอะเรย์
DISPLAY MEMORY LIKE multiply && แสดงผลลัพธ์
PROCEDURE product
PARAMETER localarray
localarray(3) = localarray(1)      * localarray(2)
RETURN
```

รูปที่ 5 ตัวอย่างโปรแกรมแสดงการผ่านค่าอะเรย์แบบอ้างอิง

ชื่ออะเรย์เพื่อผ่านค่าไปยังฟังก์ชัน ซึ่งทำให้สะดวกกว่าเดิมมากเลยทีเดียว

คำสั่ง : ช่างรู้ใจไปหมด

เมื่อเปิดดูคู่มือ FoxPro Commands & Functions แล้วก็ต้องยอมรับว่าฟอกซ์โปร 2.0 อุดมสมบูรณ์ไปด้วยคำสั่งจริง ๆ แต่การที่จะนำเอาคำสั่งทั้งหมดของฟอกซ์โปรมาอธิบายในที่นี้ก็เห็นว่าไม่เหมาะสมด้วยสาเหตุหลายประการจริง ๆ เพราะว่าจะทำให้ท่านที่ไม่เคยใช้ฟอกซ์โปรเกิดไม่เข้าใจ ด้วยยังไม่คุ้นเคยและอาจจะกลายเป็นเรื่องยากเกินไป อาจจะทำให้เบื่อก็ได้ แต่ก็พอจะสรุปได้ว่า คำสั่งทั้งหมดของฟอกซ์โปรสามารถแบ่งกลุ่มออกเป็น 6 กลุ่มและในแต่ละกลุ่มยังมีประเภทย่อย ๆ ลงไปอีก ซึ่งนับรวมกันได้ถึง 25 ประเภท โดยในเวอร์ชันนี้มีการเพิ่มและปรับปรุงคำสั่งและฟังก์ชันขึ้นอีกมากมาย (มากกว่า 100 คำสั่ง) โดยได้คัดเอาจุดที่เป็นหัวกะทิจริง ๆ นำมาเสนอไว้ดังนี้

การรับข้อมูล/แสดงผลข้อมูล

ประกอบด้วยคำสั่ง @...SAY/GET, @...GET, @...EDIT และคำสั่ง READ ในกลุ่มนี้จัดว่าพัฒนาขึ้นมามากทีเดียว โดยเฉพาะในคำสั่ง @...GET ได้จัดเตรียมให้เราสามารถสร้างองค์ประกอบในการรับข้อมูลได้หลายรูปแบบเช่น Check Boxex, Invisible Buttons, Lists, Popups, Push Buttons, Radio Buttons ดังตัวอย่างในรูปที่ 6 และคำสั่ง @...EDIT ก็ช่วยให้เราสามารถรับข้อมูลที่เป็นข้อความที่มีความยาวไม่จำกัดได้อีกด้วย ส่วนทางด้านคำสั่งของ READ ก็จะมีสีสันเพิ่มขึ้นเหมือนกัน เป็นต้นว่า

รูปที่ 6 การรับข้อมูล/แสดงผลข้อมูล

```
***** GET CHECK BOX **********************
***** FUNCTION and PICTURE clause together
STORE 1 TO choice
@ 4,2 GET choice FUNCTION '*C' PICTURE ' Tax Exempt'
READ

***** GET INVISIBEL BUTTON ****************
***** FUNCTION and PICTURE clauses together
STORE 1 TO choice
@ 4,2 GET choice FUNCTION '*I' PICTURE ' ;;;;' SIZE 2,4,1
READ

***** GET POPUP ***************************
***** FUNCTION and PICTURE clauses together
STORE 1 TO choice
@ 4,2 GET choice FUNCTION '^' ;
PICTURE ' Cash;Charge;Net 30;Net 60'
READ

***** GET PUSH BUTTON *********************
***** FUNCTION clause only
STORE 1 TO choice
STORE '* OK;Cancel' TO mprompts
@ 2,2 GET choice FUNCTION mprompts SIZE 1, 8
READ

***** PICTURE clause only
STORE 1 TO choice
@ 2,2 GET choice PICTURE '@* OK;Cancel' SIZE 1, 8
READ

***** GET RADIO BUTTON ********************
***** FUNCTION and PICTURE clause together
STORE 1 TO choice
@ 4,2 GET choice FUNCTION '*R' ;
PICTURE ' Apples;Apricots;Lemons;Oranges'
```

รูปที่ 6 (ต่อ)

```
READ

***** @ ... EDIT ************************
***** Editing region with right justified text
STORE 'This will be edited' TO text
@ 4,2 EDIT text FUNCTION 'J' SIZE 5,50
READ

***** Override color defaults
STORE 'To be edited' TO text
@ 5,0 EDIT text COLOR SCHEME 4 SIZE 5,50
READ
```

สามารถกำหนดสีสำหรับฟิลด์ที่กำลังรับข้อมูลให้ต่างจากฟิลด์อื่น ๆ ได้ และสามารถเชื่อมโยงการทำงานกับวินโดว์ได้เป็นอย่างดี และเรายังสามารถใช้คำสั่ง READ ซ้อนกันได้ถึง 5 ระดับ และยังมีคำสั่ง CLEAR READ เพื่อหยุดการทำงานของการรับข้อมูลได้ทันทีที่ต้องการอีกด้วย

การจัดการเมนู

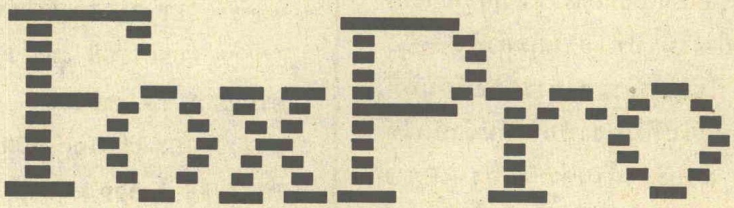
ในเรื่องนี้ ฟอกซ์โปรออกจะเด่นมากเลยทีเดียว คุณสามารถสร้างเมนูได้ (ใกล้เคียง) กับนักโปรแกรมระดับมืออาชีพอย่างไม่ต้องออกแรงคิดมากมาย เพราะได้มีคำสั่งอำนวยความสะดวกและมีทางเลือกไว้อย่างครบครันด้วยคำสั่ง DEFINE MENU สำหรับเมนูบาร์ และ DEFINE POPUP คุณยังสามารถกำหนดคีย์ฉับไว (Hot key) สำหรับการกระตุ้นป๊อบอัพและเมนูบาร์ได้อีกด้วย โดยเฉพาะป๊อบอัพ ออกจะหรูหรามากเอาการเลยทีเดียว คุณสามารถกำหนดให้ป๊อบอัพดึงข้อมูลจากแฟ้มข้อมูลขึ้นมาแสดงได้อย่างง่ายดาย และคุณยังสามารถเคลื่อนย้ายป๊อบอัพไปยังตำแหน่งต่างบนจอภาพได้ด้วยคำสั่ง MOVE POPUP อีกด้วย

การจัดการวินโดว์

วินโดว์ในฟอกซ์โปรเป็นส่วนที่มาช่วยให้คุณสามารถทลายกำแพงการแสดงผลข้อมูลขนาด 25×80 (พื้นที่ของจอภาพ) ได้อย่างเด็ดขาด เพราะว่าคุณจะมีพื้นที่สำหรับการแสดงผลได้ตามจำนวนหน่วยความจำที่คุณมีอยู่ด้วย การแสดงผลข้อมูลบนวินโดว์ ต้องบอกก่อนนะครับว่า เรื่องของวินโดว์ในฟอกซ์โปร ตอนนี้ยังไม่เกี่ยวข้องกับไมโครซอฟต์วินโดว์ที่กำลังฮิตกันขนาดหนักแต่อย่างใด เพราะว่าวินโดว์ในฟอกซ์โปรยังเป็นวินโดว์เชิงอักษรอยู่ (Text Oriented Window) แต่ในอนาคตก็ไม่แน่เหมือนกัน เพราะตอนนี้พฤติกรรมของวินโดว์ในฟอกซ์โปรก็คล้ายคลึงกับวินโดว์ในไมโครซอฟต์วินโดว์พอสมควรแล้ว เป็นต้นว่า คุณสามารถกำหนดวินโดว์ให้มีขนาดต่าง ๆ ตามที่ต้องการได้ (คำสั่ง DEFINE WINDOW) เคลื่อนย้ายวินโดว์ไปยังตำแหน่งต่าง ๆ ได้ (คำสั่ง MOVE WINDOW) ทำการซ่อนและแสดงวินโดว์ (HIDE/SHOW WINDOW) ทำการขยายขนาดของวินโดว์ (คำสั่ง ZOOM WINDOW ซึ่งแต่ก่อนนี้เราสามารถขยายวินโดว์ได้เหมือนกันแต่ไม่สะดวกนัก) และยังสามารถจอดวินโดว์ (docked Window) ไว้ให้เห็นเฉพาะส่วนที่เป็นชื่อวินโดว์เท่านั้น (คล้าย ๆ กับ Icon ในไมโครซอฟต์วินโดว์) และยังได้ข่าวแว่ว ๆ อย่างไม่เป็นทางการมาว่าขณะนี้กำลังมีการพัฒนาฟอกซ์โปรสำหรับไมโครซอฟต์วินโดว์อยู่ด้วย ยังไงก็อดใจรอข่าวคราวก็แล้วกันนะครับ

การนำเข้า/ส่งออกข้อมูล

จุดที่เห็นว่าน่าสนใจได้แก่ การนำเข้า/ส่งออกข้อมูล ด้วยคำสั่ง IMPORT/EXPORT ซึ่งจะช่วยให้คุณสามารถนำเอาแฟ้มข้อมูลจากโปรแกรมอื่น ๆ ที่คุณใช้อยู่ ตัวอย่างเช่นแฟ้มที่สร้างในโปรแกรม Lotus 1-2-3 Version 1-A, 2.X และ 3.X, Symphony, Frame-



FoxPro 2.0

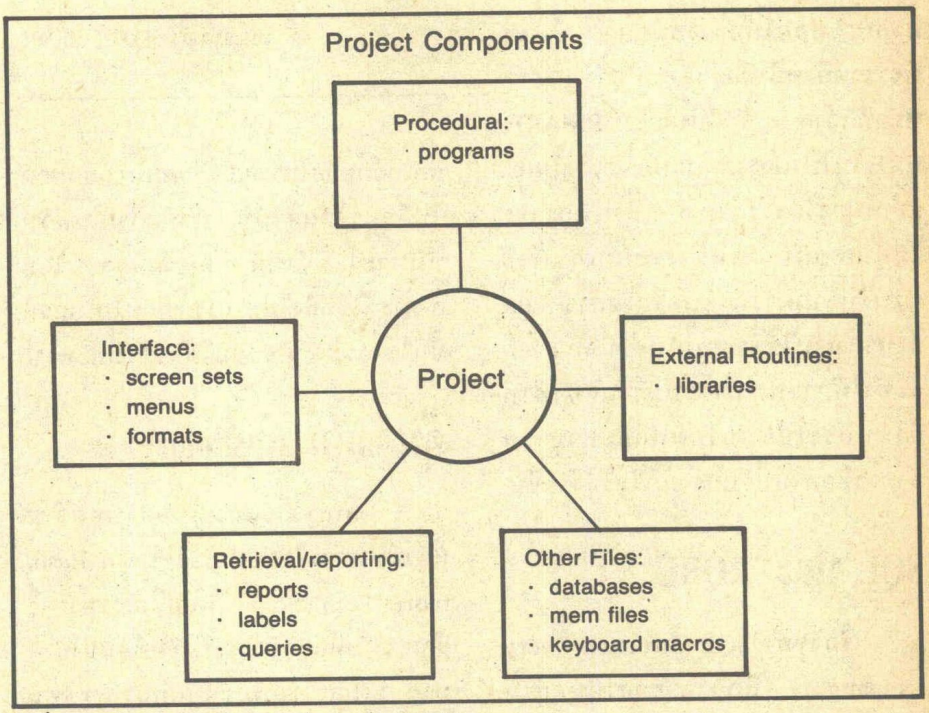
Work II, Microsoft Multiplan, Excel Version 2 PARADOX, แฟ้มข้อมูลแบบ DIF (Data Interchange Format), แฟ้มข้อมูลแบบ SYLK (Symbolic Link Interchange Format) นำมาใช้กับฟอกซ์-โปรได้ และยังสามารถนำเอาข้อมูลจากฟอกซ์โปรไปใช้ในโปรแกรมที่มีรูปแบบดังกล่าวมาแล้วได้อีกด้วย

อาวุธครบมือ

ในฟอกซ์โปร 2.0 ได้จัดเตรียมเครื่องมือเครื่องใช้สำหรับการพัฒนาระบบงานเอาไว้อย่างพร้อมมูลทีเดียว ครับจะขอรวบรัดนำเสนอได้ดังนี้

– Screen Builder ตั้งแต่ฟอกซ์-เบสพลัส 2.1 และฟอกซ์โปร 1.XX เราสามารถสร้างส่วนรับ/แสดงผลข้อมูลทางจอภาพได้โดยการใช้โปรแกรมฟอกซ์วิว แต่ว่าในฟอกซ์โปร 2.0 นี้ โปรแกรมฟอกซ์วิวได้หายไปแต่ว่ามีส่วนของ Screen Builder มาแทน ซึ่งมีความสามารถสูงกว่าเดิมมากเลย คุณสามารถสร้างองค์ประกอบต่าง ๆ ได้อย่างเหมือนกับการเขียนโปรแกรม (โดยไม่ต้องเขียนโปรแกรม) ซึ่งจะช่วยให้คุณประหยัดเวลาในการพัฒนาระบบงานลงไปมากทีเดียว และคุณยังสามารถเปลี่ยนแปลงรูปแบบของแบบฟอร์ม (บนจอภาพ) ได้ตามความพอใจ โดยไม่เสียเวลามากอีกด้วย

– Menu Builder ตรงจุดนี้ก็ช่วยผ่อนแรงสมองของคุณได้อีกเหมือนกัน คุณสามารถสร้างเมนูได้ทุกรูปแบบ อย่างที่ฟอกซ์โปรทำได้ ไม่ว่าจะเป็น Pad Menu, Popup Menu นอกจากการสร้างและกำหนดรายการต่าง ๆ ในเมนูแล้ว คุณยังสามารถเพิ่มโปรแกรม ในส่วนที่เป็นการจัดตั้งค่าต่าง ๆ ของระบบงานของคุณเป็นต้นว่า การจัดตั้งสภาวะแวดล้อมต่าง ๆ, การเปิดแฟ้มข้อมูลให้รวมอยู่ในในไฟล์เมนู (.MNX) ได้อีกด้วย



รูปที่ 7 ความสัมพันธ์ระหว่างโปรเจกต์และองค์ประกอบต่าง ๆ

– Report Writer ในฟอกซ์โปร 1.XX โปรแกรมสร้างรูปแบบรายงานก็จัดว่ามีความสามารถน่าพอใจอยู่ทีเดียว แต่ว่ายังติดขัดอยู่ในเรื่องของ Printer Driver ซึ่งก็ได้รับการแก้ไขแล้วในเวอร์ชันนี้ ซึ่งทำให้เราสามารถทำการพิมพ์ตัวอักษรได้หลายรูปแบบ ตัวอย่างเช่น ตัวเข้ม ตัวเอียง ตัวขยาย เป็นต้น (ในส่วนที่เป็น LABEL Designer ก็สามารถทำได้เช่นกัน) ผลลัพธ์ (รายงาน) ที่ได้ก็ดูมีชีวิตชีวาเพิ่มขึ้น และในด้านนำเสนอข้อมูลในรายงาน เรายังสามารถจัดกลุ่มของข้อมูลได้ถึง 20 กลุ่ม และในรายงานเราสร้างฟิลด์ขึ้นใหม่ได้ ซึ่งโดยมากแล้วจะเป็นฟิลด์ที่เกิดขึ้นจากการคำนวณ นอกจากนี้แล้วเรายังสามารถนำเอาฟังก์ชันที่เราเขียนขึ้นเอง มาใช้เป็นนิพจน์ในรายงานได้ด้วย

จากโปรแกรมสู่โปรเจกต์

ตอนนี้ฟอกซ์โปร 2.0 ได้ก้าวเข้าสู่การเป็นเครื่องมือสำหรับการสร้างระบบงานประยุกต์ มากกว่าการเป็น DBMP ธรรมดาเสียแล้ว เพราะจากเครื่องไม้เครื่องมือที่ได้ชี้แจงไว้ข้างต้นไปแล้ว จะเห็นว่าเราสามารถนำเอาองค์ประกอบต่าง ๆ มารวมกันแล้วสร้างเป็นระบบงานได้ และในบางจุดเราก็อาจจะเขียนโปรแกรมเพิ่มเติมเข้าไปได้ตามความต้องการและลักษณะงานที่จะปฏิบัติ โดยฟอกซ์โปรได้จัดเตรียมกระบวนการสำหรับการจัดการเรื่องนี้เอาไว้เรียบร้อยแล้ว ในส่วนของ Project Manager : PM โดยที่ PM จะทำหน้าที่ในการดูแลรักษาส่วนประกอบต่าง ๆ ของโครงการให้มีความทันสมัยเสมอ โดยในโครงการหนึ่งอาจจะมีองค์ประกอบหลายอย่างดังที่ได้แสดงแผนผังไว้ในรูปที่ 7 PM จะมีกรรมวิธีในการติดตามและรวบรวม

องค์ประกอบให้อย่างสมบูรณ์ ซึ่งเมื่อพิจารณาถึงหน้าที่แล้วก็จะเห็นว่า PM ทำหน้าที่คล้าย ๆ กับโปรแกรม RMAKE.EXE ในโปรแกรมคลิปเปอร์ 5.0 นั่นเอง แต่เห็นว่าในระยะแรก ๆ นี้ท่านอาจจะไม่คุ้นเคยกับการสร้างระบบงานด้วยการเริ่มจากการกำหนดโครงการ แต่ถ้าหากท่านใช้ขั้นตอนเหล่านี้แล้ว ก็จะทำให้การพัฒนาระบบงานของท่านมีระบบระเบียบมากขึ้นและสามารถทำการติดตามปรับปรุงแก้ไขได้ง่ายขึ้น

SQL และ RQBE

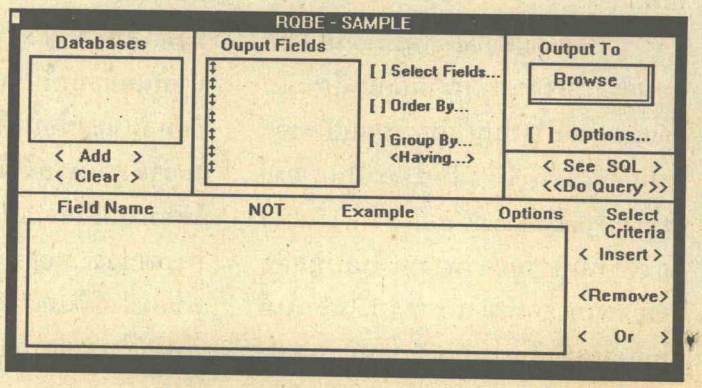
เรื่องของ SQL (Structure Query Langauge) รู้สึกว่าจะฮือฮากันอยู่ในช่วงที่ dBASE IV ออกตัวมาใหม่ ๆ แต่ทว่าเดี๋ยวนี้ก็เงียบไปแล้ว ซึ่งอาจจะเนื่องจากว่า SQL ใน dBASE IV นั้นไม่ค่อยสมบูรณ์และยังใช้งานได้ไม่ค่อยสะดวกนัก ในฟอกซ์โปร 2.0 ก็มีคำสั่งพื้นฐานของ SQL เหมือนกัน ซึ่งได้แก่ คำสั่ง SELECT และจะเห็นว่าคำสั่ง SELECT ใน SQL ต่างจากคำสั่ง SELECT ที่ใช้เลือกพื้นที่ใช้งานโดยสิ้นเชิง ดังรูปแบบที่แสดงไว้ในรูปที่ 8 แต่ทว่าตอนนี้ต้องของดกล่าวในเรื่องของรายละเอียดการใช้งานเอาไว้ก่อนแล้วจะนำเสนอเป็นบทความในแง่การใช้งานอีกครั้งหนึ่ง นอกจากนี้แล้วฟอกซ์โปรยังได้จัดเตรียมส่วนที่เป็น RQBE (Relational Query By Example) ซึ่งจุดนี้จะช่วยให้ผู้ใช้ที่ไม่สามารถเขียนโปรแกรมได้ใช้งานฟอกซ์โปรได้อย่างมีประสิทธิภาพ เพราะคุณสามารถใช้ RQBE Window ดังในรูปที่ 9 ทำการสอบถามข้อมูลที่ต้องการได้โดยปราศจากการเขียนโปรแกรม และยังเป็นการใช้ประโยชน์ในส่วนของ Rushmore Technology อย่างเต็มที่อีกด้วยในรายละเอียดผมขอยกยอดไปอีกครั้งก็แล้วกัน แต่คิดว่าเป็นส่วนที่มีประโยชน์มาก ๆ เลยทีเดียว โดยเฉพาะผู้ใช้ที่ยังไม่สามารถเขียนโปรแกรมขึ้นใช้งานได้ ก็จะยิ้มได้คราวนี้ล่ะครับ

รูปที่ 8 คำสั่ง SELECT ใน SQL

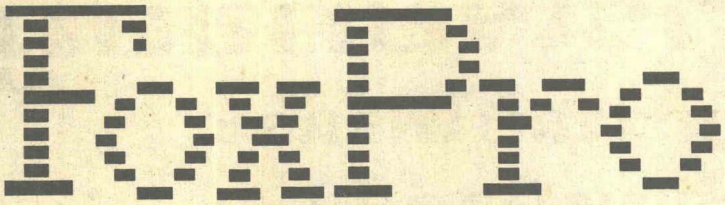
```
SELECT [ALL ¦ DISTINCT]
[<alias>.]<select_item>
[AS <column_name>]
[, [<alias>.]<select_item>
[AS <column_name>] ...]
FROM <database> [<local_alias>]
[, <database>
[<local_alias>] ...]
[[INTO <destination>]
¦ [TO FILE <file>
[ADDITIVE] ¦ TO PRINTER]]
[NOCONSOLE]
[PLAIN]
[NOWAIT]
[WHERE <joincondition>
[AND <joincondition> ...]
[AND ¦ OR
<filtercondition>
[AND ¦ OR
<filtercondition> ...]]]
[GROUP BY <groupcolumn>
[, <groupcolumn> ...]]
[HAVING <filtercondition>]
[UNION [ALL] <SELECT command>]
[ORDER BY <order_item>
[ASC ¦ DESC]
[, <order_item>
[ASC ¦ DESC]...]]
```

คุ้มค่ากับ Rushmore

ได้กล่าวไว้ตอนต้นของบทความแล้วว่า เราจะได้รับประโยชน์จาก Rushmore Technology ก็ต่อเมื่อเราทำการจัดการกับข้อมูลที่มีความซับซ้อนยุ่งยากมาก ๆ (และใช้แฟ้มข้อมูลหลาย ๆ ไฟล์ร่วมกัน) แต่อาจจะมีคำถามว่า ถ้าหากว่าเราใช้แฟ้มข้อมูลเพียงแฟ้มเดียว เราจะได้รับประโยชน์จากเทคนิคอันนี้ได้อย่างไร คำตอบก็มีอยู่ว่า เราสามารถได้รับผลพวงของเทคนิคอันนี้เมื่อเราใช้คำสั่งที่มีอนุประโยค FOR ในการสร้างเงื่อนไข ซึ่งจะช่วยเพิ่มความเร็วของการดึงข้อมูล แต่ทว่าคุณจะต้องทำตามกติกาในการสร้างนิพจน์เงื่อนไขให้ถูกต้อง (ตามที่ระบุเอาไว้ในคู่มือ Developer's Guide) ด้วยนะครับ ไม่เช่นนั้นแล้วคุณก็อาจจะไม่ได้ผลลัพธ์ที่น่าพอใจก็ได้ และมีข้อควรระวังอีกอย่างคือถ้าหากคุณใช้อนุประโยค WHILE ในคำสั่ง ก็



รูปที่ 9 RQBE WINDOW



จะเป็นการปิดการทำงานของ Rushmore ไปทันที และในฟอกซ์โปร Standard Version จะใช้ความสามารถของเทคนิคอันนี้ได้ก็ต่อเมื่อข้อมูลที่คุณจัดการอยู่มีไม่เกิน 500,000 รายการ แต่ใน Extend Version ไม่ได้ระบุข้อจำกัดเอาไว้

เมื่อใช้หลายคน

ในที่นี้ขอกล่าวถึงในแง่ของการใช้งานในระบบเครือข่ายท้องถิ่น (LAN : Local Area Network) โดยตรง แต่เดิมแล้ว DBMP ตระกูลฟอกซ์ จัดได้ว่ามีความคล่องตัวในการใช้งานในระบบมากพอตัวทีเดียวในฟอกซ์โปร 2.0 นี้ก็เหมือนกัน สำหรับระบบแลนที่สามารถใช้กับฟอกซ์โปร (เท่าที่ระบุไว้ในคู่มือ) ก็ได้แก่ 3Com3+, Novell Advanced Netware (Version 1.02 ขึ้นไป), Novell Netware 286, Novell Netware 386, Lantastic, Invisible Net รู้สึกว่าฟอกซ์โปร 2.0 ได้ออกแบบให้ใกล้ชิดสนิทสนมกับระบบแลนของ Novell พอสมควร (สังเกตจากคำสั่ง SET PRINTER TO ในรูปแบบที่ 2 ซึ่งออกแบบมาสำหรับระบบ Advanced Netware โดยเฉพาะ) และขั้นตอนที่สำคัญเมื่อคุณต้องการใช้ฟอกซ์โปรบนระบบแลน หลังจากคุณได้ทำการติดตั้งฟอกซ์โปรเรียบร้อยแล้ว คุณต้องทำการรันโปรแกรม ADDUSER.PRG เพื่อสร้าง CONFIG.FP, RESOURCE FILE และ OVERLAY FILE สำหรับแต่ละเวอร์กสเตชันเสียด้วยซึ่งอาจจะมีขั้นตอนยุ่งยากขึ้นนิดหนึ่ง แต่เห็นว่าเป็นเรื่องที่ทำให้ยุ่งยากขึ้นนี่แหละจะทำให้ระบบฐานข้อมูลของคุณทำงานได้อย่างน่าเชื่อถือยิ่งขึ้น จุดที่น่าสนใจของฟอกซ์โปรในเรื่องของระบบแลน จุดแรกก็เห็นจะเป็นเรื่องของกลุ่มคำสั่งที่จัดการกับแฟ้มข้อมูลที่ทำการล็อกไฟล์และเรคอร์ดโดยอัตโนมัติ โดยถ้าหากเราพิจารณาเลือกใช้คำสั่งในกลุ่มนี้ให้เหมาะสม จะทำให้โปรแกรมที่เราเขียนขึ้นสามารถทำงานได้ดีทั้งในระบบที่เป็น Single User Mode และ Multi User Mode และยังมีคำสั่ง SET MULTILOCK ON : OFF ที่ทำการล็อกกลุ่มของเรคอร์ดได้อีกด้วย และคำสั่ง SET REFRESH TO <ตัวเลข> และ SET REPROCESS TO <ตัวเลข> ก็เป็นคำสั่งที่ช่วยให้การทำงานบนระบบแลน น่าเชื่อถือยิ่งขึ้นอีกด้วย แต่อย่างไรก็ตาม รายละเอียดในเรื่องของการใช้งานฟอกซ์โปรในระบบแลนก็คงเป็นเรื่องที่ต้องนำมากล่าวกันอีกครั้งหนึ่ง ซึ่งคิดว่าจะนำมาขยายให้ท่านทราบในเร็ว ๆ นี้

ยังไม่ได้กล่าวถึง

ส่วนที่ผู้เขียนยังไม่ได้นำมาแจกแจงให้ท่านทราบ มิใช่เป็นเจตนาที่จะปกปิด แต่เห็นว่าเป็นเรื่องที่มีความสำคัญน้อยลงไปกว่าเรื่องราวที่นำเสนอมาแล้ว และอีกอย่างก็เกรงว่าบทความจะยาวเกินไป สำหรับเรื่องราวที่ยังไม่ได้กล่าวถึงได้แก่ การปรับแต่งค่าต่าง ๆ ในฟอกซ์โปร, การใช้แฟ้มแบบ Low-Level, การสร้างเอกสารประกอบระบบงานด้วย FoxDoc, เรื่องของ Text Merge กับภาษาร่างแบบ, การใช้งานกระบวนการ Debug และ Trace ในการติดตามและแก้ไขโปรแกรม, การปรับแต่งและการใช้งานระบบให้คำแนะนำ (Help Facility), เรื่องของความเข้ากันได้ (Compatibility) ระหว่างฟอกซ์โปร 1.XX, ฟอกซ์เบสพลัสกับฟอกซ์โปร 2.0 โดยเรื่องเหล่านี้คิดว่าคงต้องให้ผู้ที่ต้องการใช้ฟอกซ์โปรไปศึกษาในรายละเอียดเอาเอง

สรุป

เมื่อเรามาจนถึงตรงนี้แล้ว อยากจะเสนอข้อคิดสักนิดหนึ่งว่าถ้าคุณเป็น "คนเขียนโปรแกรม" และคุณคิดจะร่วมวงไพบูลย์กับฟอกซ์โปร โดยที่ยังไม่เคยใช้มาก่อนเลยล่ะก็ ผมอยากแนะนำให้คุณเริ่มต้นที่ ฟอกซ์โปร 1.XX เพราะว่า จะทำให้คุณสามารถทำความเข้าใจกับสิ่งต่าง ๆ ได้ง่ายกว่า และจะสามารถทำได้ดีขึ้นในฟอกซ์โปร 2.0 แต่ถ้าหากคุณเคยใช้ฟอกซ์โปรมาบ้างแล้ว ผมคิดว่าคุณก็น่าจะตบเกียร์เดินหน้าได้เลย และถ้าหากคุณเป็น "ผู้ใช้" ผมคิดว่าคุณน่าจะกระโดดเข้าหาฟอกซ์โปร 2.0 ได้เลย เพราะว่าฟอกซ์โปร 2.0 จะช่วยให้คุณสามารถสร้างงานได้อย่างน่าพอใจ โดยอาจจะกล่าวได้ว่า ฟอกซ์โปร 2.0 จะพาคุณก้าวเข้าสู่ การสร้างโปรแกรมโดยผู้ใช้งานได้เป็นอย่างดีทีเดียว แต่ถึงอย่างไร เรื่องของการตัดสินใจเลือกและไม่เลือก หรือว่าจะเริ่มยังไงตรงไหน (แต่อย่าตัดสินใจช้านักนะครับ) ก็ขอยกหน้าที่อันสำคัญนี้ให้คุณพิจารณาก็แล้วกัน แล้วค่อยเจอกันใหม่นะครับ